

Amendment to the Claims

Kindly amend claims 1, 27, 53 & 58 and cancel claims 19, 20, 45, 46, 76 & 77 without prejudice as set forth below. In compliance with the Revised Amendment Format published in the Official Gazette on February 25, 2003, a complete listing of claims is provided herein. The changes in the amended claims are shown by strikethrough (for deleted matter) and underlining (for added matter).

1. (Currently Amended) A method of managing thread pools of a computing environment, said method comprising:

receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of ~~one or more~~ eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said dynamically altering comprises setting a pool mask ~~associated with said response~~ to indicate said ~~one or more~~ eligible thread pools of said altered thread pool set to service said response.

2. Canceled

3. (Previously Presented) The method of claim 1, wherein said pool mask is included within a data structure associated with said response.

4. (Original) The method of claim 1, wherein said dynamically altering is initiated when it is determined that said request is waiting for said response.


5. Canceled

6. Canceled

7. Canceled

8. Canceled

9. (Original) The method of claim 1, further comprising dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

 10. (Original) The method of claim 9, wherein said dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

11. (Original) The method of claim 10, further comprising determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

12. Canceled

13. Canceled

14. (Original) The method of claim 1, wherein said receiving comprises receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

15. (Original) The method of claim 14, wherein said server is a file server.

16. (Original) The method of claim 14, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

17. (Original) The method of claim 14, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

18. (Original) The method of claim 1, wherein said dynamically altering is performed by a server of said computing environment.

19. Canceled

20. Canceled

21. (Original) The method of claim 1, wherein said first requester and said second requester are the same requester.

22. (Original) The method of claim 1, wherein said first requester and said second requester are different requesters.

23. Canceled

24. Canceled

25. Canceled

26. Canceled

27. (Currently Amended) A system of managing thread pools of a computing environment, said system comprising:

means for receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

means for, upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of ~~one or more~~ eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said means for dynamically altering comprises means for setting a pool mask ~~associated with said response~~ to indicate said ~~one or more~~ eligible thread pools of said altered thread pool set to service said response.

28. Canceled

29. (Previously Presented) The system of claim 27, wherein said pool mask is included within a data structure associated with said response.

30. (Original) The system of claim 27, wherein the dynamically altering is initiated when it is determined that said request is waiting for said response.

31. Canceled

32. Canceled

33. Canceled

34. Canceled

35. (Original) The system of claim 27, further comprising means for dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

36. (Original) The system of claim 35, wherein said means for dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

37. (Original) The system of claim 36, further comprising means for determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

38. Canceled

39. Canceled

40. (Original) The system of claim 27, wherein said means for receiving comprises means for receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

41. (Original) The system of claim 40, wherein said server is a file server.

42. (Original) The system of claim 40, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

43. (Original) The system of claim 40, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

44. (Original) The system of claim 27, wherein the dynamically altering is performed by a server of said computing environment.

45. Canceled

46. Canceled

47. (Original) The system of claim 27, wherein said first requester and said second requester are the same requester.

48. (Original) The system of claim 27, wherein said first requester and said second requester are different requesters.

49. Canceled

50. Canceled

51. Canceled

52. Canceled

53. (Currently Amended) A system of managing thread pools of a computing environment, said system comprising:

a processor adapted to receive from a first client of said computing environment a request to be processed, said request waiting on a response from a second client of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

said processor being adapted to, upon receipt of the request waiting on the response, and without input from said first client or said second client of which thread pools can service the response, dynamically alter said set of one or more eligible thread pools to provide an altered thread pool set of ~~one or more~~ eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

said processor being adapted to set a pool mask ~~associated with said response~~ to indicate said ~~one or more~~ eligible thread pools of said altered thread pool set to service said response.

54. (Original) The system of claim 53, wherein said processor comprises a server of said computing environment.

55. (Original) The system of claim 53, wherein said first client and said second client are the same client.

56. (Original) The system of claim 53, wherein said first client and said second client are different clients.

57. Canceled

58. (Currently Amended) At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of managing thread pools of a computing environment, said method comprising:

receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of ~~one or more~~ eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said dynamically altering comprises setting a pool mask ~~associated with said response~~ to indicate said ~~one or more~~ eligible thread pools of said altered thread pool set to service said response.

59. Canceled

60. (Currently Amended) The at least one program storage device of claim ~~59~~ 58, wherein said pool mask is included within a data structure associated with said response.

61. (Original) The at least one program storage device of claim 58, wherein said dynamically altering is initiated when it is determined that said request is waiting for said response.

62. Canceled

63. Canceled

64. Canceled

65. Canceled

66. (Original) The at least one program storage device of claim 58, wherein said method further comprises dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

67. (Original) The at least one program storage device of claim 66, wherein said dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

68. (Original) The at least one program storage device of claim 67, wherein said method further comprises determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

69. Canceled

70. (Original) The at least one program storage device of claim 69, wherein said ordering comprises having a primary thread pool selectable before any secondary thread pool.

71. (Original) The at least one program storage device of claim 58, wherein said receiving comprises receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

72. (Original) The at least one program storage device of claim 71, wherein said server is a file server.

73. (Original) The at least one program storage device of claim 71, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

74. (Original) The at least one program storage device of claim 71, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

75. (Original) The at least one program storage device of claim 58, wherein said dynamically altering is performed by a server of said computing environment.

76. Canceled

77. Canceled

78. (Original) The at least one program storage device of claim 58, wherein said first requester and said second requester are the same requester.

79. (Original) The at least one program storage device of claim 58, wherein said first requester and said second requester are different requesters.

80. Canceled

81. Canceled

82. Canceled

83. Canceled